This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM23-0557

Symbolic Assurance Refinement

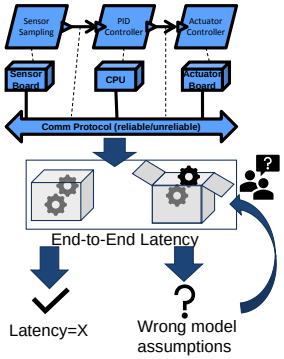
Dionisio de Niz and Lutz Wrage

Late Discovery of Design Errors in DoD Systems is very **costly**.

Architecture modeling and analysis can detect design error early

BUT:

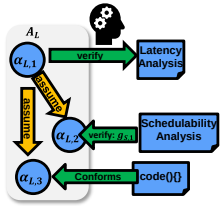
- Analysis assumptions are often implicit
- if analysis assumptions not met: analyses break down for reasons not clear to users of analysis tools.
  - E.g., e2e Latency Assumption: periods multiple of each other (harmonic)

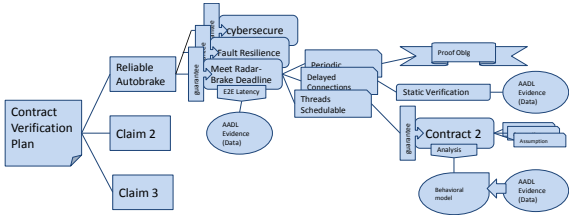


SOLUTION

```
contract {
  inputs:
                E2ELatancies
  assumptions:
                areConnectionsDelayed()
                areDeadlinesConstrained()
                areTasksSchedulable()
                areAllThreadsPeriodic()
  analysis:
                meetEndToEndLatencies()
  guarantee:
                [E2EResponses[i]  <= E2ELatencies[i]
                  for i in range(len(Responses))]
}
```

$C_L = (A_L, G_L)$ with $A_L = \{\alpha_{L,1}, \alpha_{L,2}, \alpha_{L,3}\}$



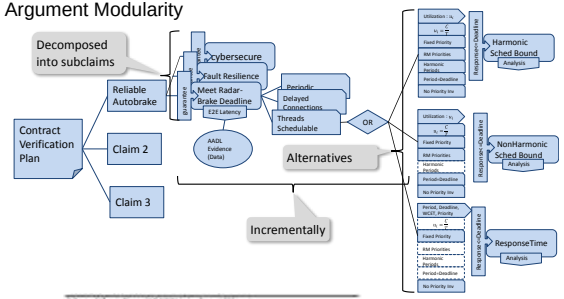


Symbolic Assurance:
- Symbolic Satisfiability Problem Encoding (SMT)

Argumentation evolves with model refinement
- Unconstrained symbolic variables assigned value to satisfy problem
- Considered as proof obligations (due to partial model)

Verify final complete model for all possible values

Argument Modularity



Refinement-Based Selection:
- select contract based on model data availability

Design-Based Selection:
- Select contract based on model data type (e.g. RM priorities)